东南大学工科试验班计算机程序设计（下）期末考核试卷

考试学期：2022/23-春　　考试形式：开卷

考试时间：2023 年 6 月 16 日 9:00 ~ 11:00

一、考试注意事项

1. 在考试系统中打开题目；
2. 在本机答题，自行建立相关工程目录与 CPP 文件；
3. 在考试结束前，务必将本机所答题文件上传到考试系统，否则不计分数；在临近考试结束前，因优先考虑上传文件，而不是继续答题；
4. 上传文件包括：（1）程序文件（.c、.cpp、.h 等，必须）；（2）运行截图（.png、.jpg 等，可选）；（3）说明文档（.docx 等，可选）；（4）运行输出的文件（如考题要求输出文件，则必须上传）；上传时注意文件选择对话框中（一般在右下角），可以选择文件类型；
5. 编译不通过的，除回答问题外，整题得分原则上不得超过本题满分的 1/2；
6. 只允许使用 VS2010 或更高版本的 IDE 环境，否则造成的任何兼容性问题而导致的扣分，由答题者自行承担；
7. 本卷题号与上传题号严格对应，不答题的题目，上传“本题不作答.txt”以视放弃；阅卷时，阅卷教师不对错传题号、误传文件等情况进行纠正，如考试系统中对应题号非对应题目，视为本题放弃作答；
8. 任何在考试结束后提交给任课教师或监考教师的代码（无论所提交文件的修改时间是否在考试期间），皆不予承认，不得作为评分依据；
9. 请务必将“代码.zip”在文件夹中解压缩，然后再答题，切勿直接双击“代码.zip”打开压缩软件答题；
10. 为提醒考生注意，请监考教师开考前 5 分钟前大声朗读本事项，但本事项不因监考教师未朗读而失效；
11. 本注意事项以答题者卷面所见为准。

二、题目设置

本试共 5 题，前 3 题每题 35 分，选 2 题作答，后 2 题每题 30 分，选 1 题作答。本卷满分 100 分。多做的题目，以最高得分的两题计分，不设附加分。

三、试题

（见下页）

第 1 题（1、2、3 题三选二，35 分）

这是二维图形操作中的一个部分：点和折线的操作。

在 A1.cpp 中，Point 类表示一个二维平面中的一个点。PolyLine 表示一条折线，成员变量 pLine 指向一个长度为 size 的数组。系统整体数据结构图如图 1：

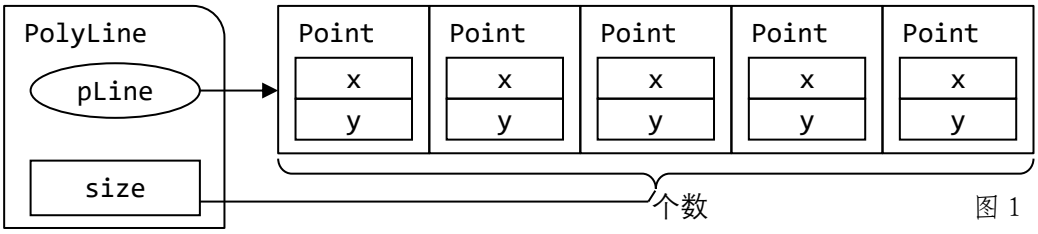

图 1

要求完成：

（1）完成 Point 和 PolyLine 类输出“<<”的重载，Point 类的“<<”重载采用友元，而 PolyLine 类的“<<”重载禁止采用友元（需自行添加一些代码）；

（2）当与人工智能大语言模型谈及友元的话题时，有以下问答（图 2）：

图 2

借此对话，说一说你在实现 PolyLine 类的“<<”非友元重载时做的工作？和对话中的描述相匹配吗？这里应当采用通用方案，而非为了运行通过的临时方案。

（3）完成 PolyLine 类的“[]”运算符重载，按索引获得折线中的点；

（4）完成 PolyLine 类的“+=”运算符重载，加入新点，注意数组空间问题。

最终要求运行主函数通过，屏幕输出所有点的坐标。

第 2 题（1、2、3 题三选二，35 分）

在战争时期，重要信息加入大量素数进行加密，哪怕被敌方截获也无法破解获得真实情报。质数的一项主要应用是在密码学中。计算两个超大质数的乘积是容易的，但是要想知道一个大整数，如 9999998660000000273 的质因子是 999999929 和 999999937 就有些困难了，这一点可以被充分利用到加密算法中。即使是在计算机的时代，只需要采用更大的质数 P1、P2，得到乘积 A，那么对于一个不知道任何信息的外部人员来说，想要对 A 进行质因数分解也是相当困难的，重点是数学界也没有找到对极大数的进行快速质因数分解的算法。

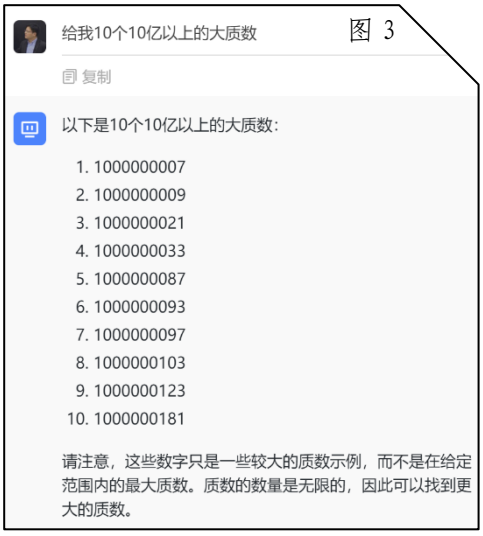

图 3

通过求助人工智能语言模型，给出了 10 个 10 亿以上的大质数（如图 3）。

A2.cpp 中提供了一个密码器类“CipherMachine”，其中“Decrypt”函数用于解密，即输入一个 long long 类型的大整数，采用 CipherMachine 内部存储的大质数逐一尝试，直到找到能将这个大整数分解的质数，并在屏幕上以“A = P1 * P2”的形式输出。

CipherMachine 类中提供了一个类似于链表的结构存储大质数，结构如图 4：

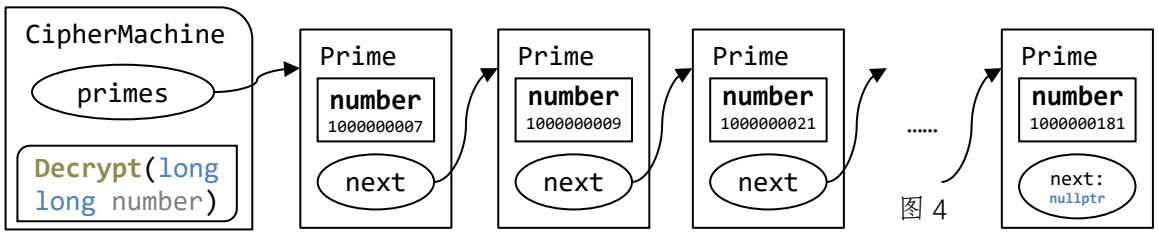

图 4

其中，Prime 类是链表的每一个节点，其中包含了 number 和 next 两个成员变量，number 中存储质数，next 指向下一个 Prime 对象。注意最后一个 Prime 对象的 next 内存储的是空指针 nullptr，表示链表结束。

要求完成：

（1）在 CipherMachine 的构造函数中，构造上图中的链表；

（2）在 Decrypt 函数中，对链表进行遍历，找到能够进行质因子分解的数字，返回该数字，若找不到，返回-1；

（3）主函数调用 Decrypt 函数，且在屏幕上以“A = P1 * P2”的形式输出（A、P1、P2 替换为具体的数字），请将“A = P1 * P2”的形式（A、P1、P2 替换为具体的数字）写入文件“Decrypt_学号.txt”并上传到考试系统。

（4）回答问题：如果要对 10000 位的大整数进行质因子分解，每个质因子可能高达 5000 位，从整数表达上，你有什么好的设想或建议？

第 3 题（1、2、3 题三选二，35 分）

方程求解是工程领域的重要问题，但很多方程尤其是非线性方程通常不存在解析解，因此探索方程的快速数值解法意义重大。牛顿迭代法（Newton's method）又称为牛顿-拉夫逊（拉弗森）方法（Newton-Raphson method），是牛顿在 17 世纪提出的一种在实数域和复数域上近似求解方程根的方法。

依据牛顿迭代法，方程$f(x)$=0 在 x_0 附近的根（即函数$f(x)$与x轴的交点）可以通过如下公式求取：

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

其中$f'(x_n)$表示函数$f(x)$的一阶微分$f'(x)$在第 n 次迭代值附近的导数值，x_0由用户指定，此后x_1可由x_0递推，后x_2可由x_1递推，以此类推，如果第i次迭代满足$|x_i - x_{i-1}|$小于事先规定的阈值，则停止迭代，且认为x_i就是方程的根，右图表达了该迭代过程。

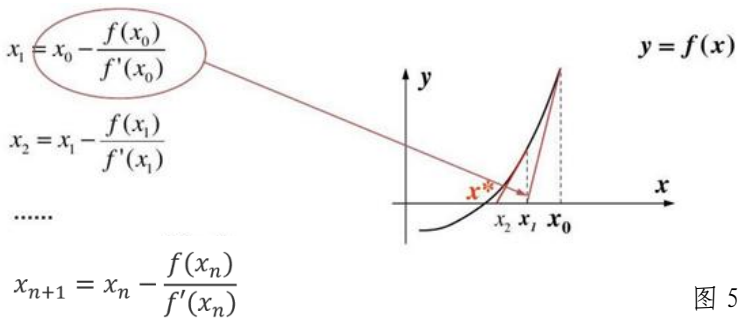

图 5

方程求解器 FunctionResolver 类支持对多项式方程求解。GetRoot 函数代表求解，多项式函数为 F()、多项式导数函数为 Diff()。FunctionResolver 类中的数组 arr 记录了多项式系数，arrSize 记录了系数的个数。arr 数组中，对应 x 的次方与数组下标一致，且各类关系如表 1 所示（以$f(x) = 4 - x + 5x^2 + x^3$为例）：

数组内容	4	-1	5	1
数组索引	0	1	2	3
对应x次方	x^0	x^1	x^2	x^3
对应多项式系数	4	-1	5	1
对应求导x次方	无	x^0	x^1	x^2
对应求导系数	0	-1×1	5×2	1×3
		arrSize 个		

表 1

要求完成

（1）依据上表，完成函数 F、Diff；

（2）利用牛顿迭代完成函数 GetRoot；

（3）完成主函数中两个方程求解的测试；

（4）回答问题：如果要设计一个通用方程求解器，适合更多的方程，你会如何考虑？利用本学年所学，说说你的设想。

第 4 题（4、5 题二选一，30 分）

对集合的处理是数据库、人工智能底层算法中非常重要的基础单元。

这是一个基于线性表与模板，实现一个通用性的集合类，完成集合数据插入、删除、赋值等基础功能。在 A4.cpp 中，Collection 类结构如下：

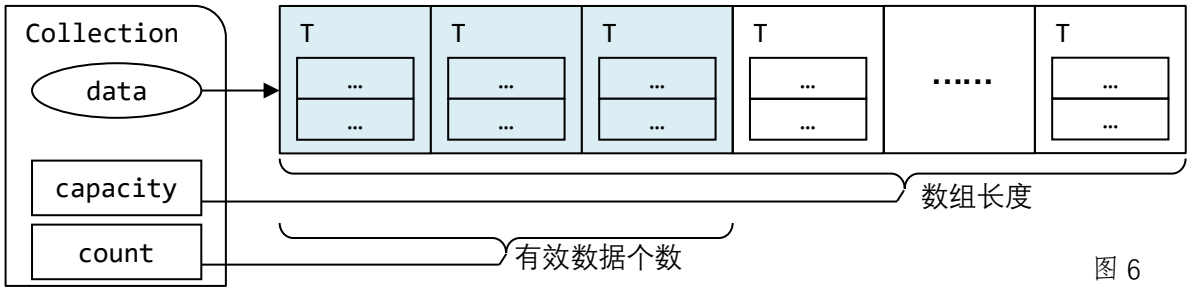

图 6

其中，data 指向了一个 T 类型的数组，capacity 为数组容量，即数组长度，count 代表了数组中有效数据的个数。

A4.cpp 中已经完成了对该集合的添加、删除等工作，请继续完成：

（1）完成 Sort 成员函数，对集合进行排序；参数 desc 为 false 时从小到大排序，参数 desc 为 true 时，从大到小排序。

（2）类 Fraction 表示一个分数，能够使用“<<”输出，但是没法参与 Collection 排序，请对 Fraction 进行适当改造，允许 Fraction 参与到 Collection 的排序中来，要求这种解决方案必须是符合工程规范的，能够适用于更广泛的场合，而非“临时性”的。

（3）回答问题：这个集合类的数据结构有什么缺点？如何改进？

第 5 题（4、5 题二选一，30 分）

在机器学习领域，卷积是非常重要的特征提取工具。不同的卷积核将能够提取信号中完全不同的特征。

如下图，卷积开始，卷积核与信号对齐在开头 0 的位置，对应位置相乘并求和得到 11。卷积核在信号上向后平移 1 个位置，再次将对应位置相乘并求和得 8，就这样不断平移，就得到了新的卷积序列：

$$c_j = \sum_{i=0}^{n-1} k_i x_{i+j}$$

其中 n 为卷积核长度，k为卷积核，x为待处理信号。

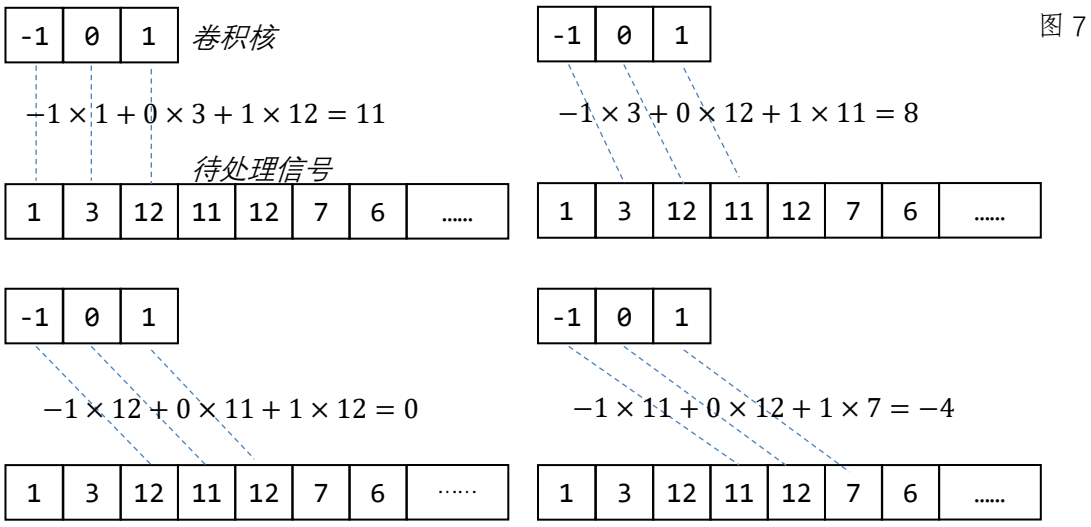

图 7

上图中卷积后的前 4 个特征为：11、8、0、-4（如下图）。不难发现这个卷积核对信号数据变大敏感；如果要检测信号在哪里突然下降，则可以使用卷积核[1, 0, -1]检测；如果要检测毛刺信号，则采用卷积核[-1, 2, -1]

Filter 类已经完成了基本的卷积处理。Filter 派生 FilterA 和 FilterB，分别实现上升沿信号和毛刺信号的检测，其卷积核分别是[-1, 0, 1]和[-1, 2, -1]：

（1）在 FilterA 和 FilterB 中，覆写了 Filter 中的纯虚函数 int GetConvCore(double* kennel, int kenelMaxSize)请将该函数补充完整；其中：kennel 指向卷积核存放内存的首地址，kenelMaxSize 代表该卷积核最大长度，该函数返回实际需要的卷积核长度。例如对于卷积核[2, 0, 1, -3]，那么只要 kenelMaxSize 大于等于 4，将 kennel[0]至 kennel[3]分别赋值为 2、0、1、-3，返回 4 即可。

（2）Filter 函数的 Conv 函数负责进行卷积，请对该函数进行完善：当卷积结果超过设定阈值，以屏幕输出的形式报警，输出“在…位置，检测到…信号”。

（3）回答问题：Filter 如果要同时采用多个卷积核卷积并输出，从数据结构，到函数设置，你会采用怎样的策略？